**Final Project**

**Group Member: Yihao Yang (group 13)**

***Abstract* – In this final project, I have finished an image classification task. The main work is to use the mage classification technique to find Penta kill frames in a video. The main process is data source finding, annotating, data preparation, training and evaluation. Finally, when I got the final working model, I found it impractical to find Penta kills in streaming videos because the Penta kill is too rare.**

1. **Introduction**

**The problem I am trying to address is to find Penta kill frames in a video. My motivation is the Penta kill is considered as one of the most exciting moments in a game. Hence, I want to use machine to extract these interesting moments and present those moment to audience on YouTube to earn views. To finish this task, I hypnosis that the model is robust enough to correctly distinguish each frame even in large number of frames due to the rarity of Penta kill. The second hypnosis I made is the views I get is worth the cost of finding these frames. Unluckily, after researched some statistics in the game of League of Legend, the Penta kill happened too rare that make the two hypnosis false.**

1. **Approach**

**The first thing I have done is to find Penta kill videos as training data source. I have luckily found an album on YouTube that mainly demonstrate Penta kill video clips. So, this album is perfect for my task here.**

**The second thing is to annotate the videos. I have totally got 15 videos. Each of them is about 1 minutes long. Then I have annotated each video into no kill frame, one to four kill frame and Penta kill frame.**

**Third, I have cropped the video frames at the upper middle part where is the Penta kill mark appears. This way makes the frame image much smaller and make the model focus on the part of image that we care.**

**Fourth, I have tried to find a good deep learning model to do classification. At a website called paperswithcode, by trade off the performance and how large the model, I have chosen the Efficient Net as my task backbone. This model has comparably very little parameters but has excellent performance. Then I tried to find some tutorials about how to use the Efficient Net using PyTorch.**

**Finally, I trained the model and try to do some demo job. I found serious problem. When I use the model to predict a video, I found lots of false positive. There are two reasons. One is the amount of training data is not enough and also the data is not diverse enough. Since I have annotated the data in a continuous manner, a significant amount of training data is similar. Also, I have only annotated about 15 15-minuts long videos. The data is not enough for a deep learning model. Another sad thing that I realized is that the Penta kill happened too rare. After some search online, I found the Penta kill happened once in 1500 games. And each game is average 35 minutes long. I have measured the time I need to process a 10 minutes video is 7 seconds. Through these statistics, I estimate that I have to take 38 hours to get a single Penta kill not including the time to download those videos. So, this project is not doable in real life.**

1. **The dataset**

**The dataset is a bunch of images which comes from game videos. The total images I have annotated is about 65k images. Some demo of the dataset is as following:**

**Penta kill frame:**



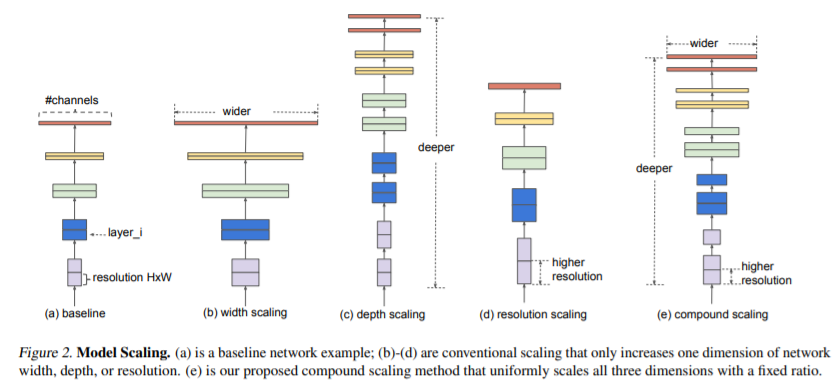
**None Penta kill frame:**



**In the whole dataset, around 60% of the data are Penta kill frames, and 40% are none Penta kill. The 80% used for training and 20% for testing.**

**When testing, I got precision: 0.95, recall: 0.99, f1 score: 0.97. The results seem pretty nice, but in this task due to the Penta kill happened too rare, the model have too look through nearly 2 million images to find a single one Penta kill. Thus, even 0.05 error will make the prediction looks bad. The only way to mitigate these is to annotate more data.**

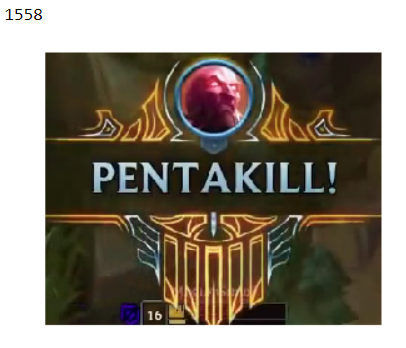
1. **Efficient Net**



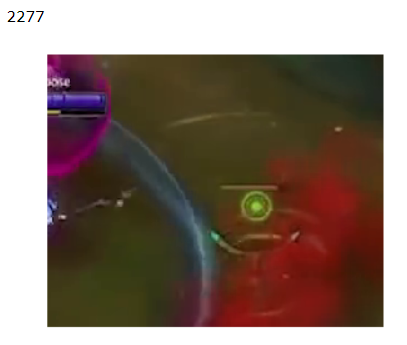
**The baseline model come from neural architecture search. Then the model is scaled up in aspects of width, depth and resolution by an efficient coefficient compound. Specifically, this project use the smallest version EfficientNet, the baseline B0.**

1. **Results**

**True positive:**



**False positive:**



**The number above the image is the frame index of the video.**